

MULTIMEDIA



UNIVERSITY

STUDENT ID NO

|  |  |  |  |  |  |  |  |  |  |
|--|--|--|--|--|--|--|--|--|--|
|  |  |  |  |  |  |  |  |  |  |
|--|--|--|--|--|--|--|--|--|--|

VENUE: \_\_\_\_\_

TABLE NO: \_\_\_\_\_

# MULTIMEDIA UNIVERSITY

## FINAL EXAMINATION

TRIMESTER 1, 2015/2016

### TCP1121 – COMPUTER PROGRAMMING

( All sections / Groups )

6 OCTOBER 2015  
2.30 p.m - 4.30 p.m  
( 2 Hours )

| Question | Mark |
|----------|------|
| 1        |      |
|          |      |
|          |      |
|          |      |
| Total    |      |

---

#### INSTRUCTION TO STUDENT

1. This Question paper consists of 14 pages with 5 Questions only.
2. **Question 1** is compulsory.
3. Attempt any **THREE** from Question 2 to Question 5.
4. Each question carries 10 marks and the distribution of the marks is given.
5. Please write all your answers in **this Question paper**.

**Question 1****All students must answer this question.**

Write the output as the result of execution of the following code segments. [10 marks]

(a) `int u = 7, v = 6, w = 9, x = 5;`  
`if (u != v) w = 1 + x--;`  
`else w--;`  
`cout << w << " " << x;`

(b) `int num1 = 2;`  
`while (num1 != 14){`  
 `cout << num1 << endl;`  
 `num1 += 3;`  
`}`

(c) `for (int i =1; i<4; i++){`  
 `int y = i % 3;`  
 `switch (y){`  
 `case 0:`  
 `cout << "Certain "<<endl;`  
 `break;`  
 `case 1:`  
 `cout << "Likely "<<endl;`  
 `break;`  
 `case 2:`  
 `cout << "Evans "<<endl;`  
 `break;`  
 `case 3:`  
 `cout << "Unlikely "<<endl;`  
 `break;`  
 `default:`  
 `cout << "Impossible "<<endl;`  
 `break;`  
 `}`  
`}`

(d) `void num_game(double d, double &e){`  
 `d *= 2;`  
 `e *= 2;`  
`}`  
`int main(){`  
 `double a = 4, b = 3.5;`  
 `num_game(a,b);`  
 `cout << a << " " << b;`  
`}`

Continued...

(e) `int num[6] = {2, 4, 3, 1, 5, 12};`  
`cout << num[1]*num[num[3]];`

(f) `class Sample{`  
    `int a;`  
    `public:`  
        `void init(){`  
            `a = 5;`  
            `cout<<a<<endl;`  
        `}`  
  
        `void init(int a){`  
            `a = 3;`  
            `cout<<this->a<<endl;`  
        `}`  
};

`int main(){`  
    `Sample a;`  
    `a.init();`  
    `a.init(10);`  
}

(g) `class Lecturer {`  
    `static int count;`  
    `public:`  
        `Lecturer() { ++count; }`  
        `~Lecturer() { --count; }`  
        `static int getCount() {`  
            `return count;`  
        `}`  
};

`int Lecturer::count = 0;`

`int main() {`  
    `Lecturer a;`  
    `{`  
        `Lecturer b;`  
        `cout << a.getCount() <<endl;`  
    `}`  
    `cout << Lecturer::getCount() <<endl;`  
}

**Continued...**

```
(h) class Lecturer {
    static int count;
    public:
        Lecturer() { ++count; }
        ~Lecturer() { --count; }
        static int getCount() {
            return count;
        }
};

int Lecturer::count = 0;

int main() {
    Lecturer a, b;
    cout << a.getCount() << endl;
    {
        Lecturer c;
        cout << c.getCount() << endl;
    }
}
```

```
(i) class X{
    public:
        virtual ~X(){ cout<<"X\n";}
};

class Y: public X{
    public:
        ~Y(){ cout<<"Y\n";}
};

int main(){
    X* x1 = new Y;
    delete x1;
}
```

```
(j) int main() {
    vector<int> nums;
    nums.insert (nums.begin(), 5);
    nums.insert (nums.begin(), 4);
    nums.insert (nums.end(), 3);
    nums.insert (nums.begin()+2, 2);

    cout << nums[1] << endl;
    cout << nums[3] << endl;
}
```

Continued...

Attempt any THREE from Question 2 to Question 5.

**Question 2**

- a. State the difference between a while loop and a do-while loop. [2 marks]

- b. The following code prompts the user to input the heights of Basketball team players and stores them in array height. The program is to be completed so it can sort the array height in descending order before displaying the array. Write the missing code in the allocated space. [3 marks]

```
int main(){
    float height[5];
    float tmp;
    for(int i=0; i<5; i++){
        cout << "Enter the height: ";
        cin >> height[i];
    }
    cout.setf(ios::fixed | ios::showpoint);
    cout.precision(1);

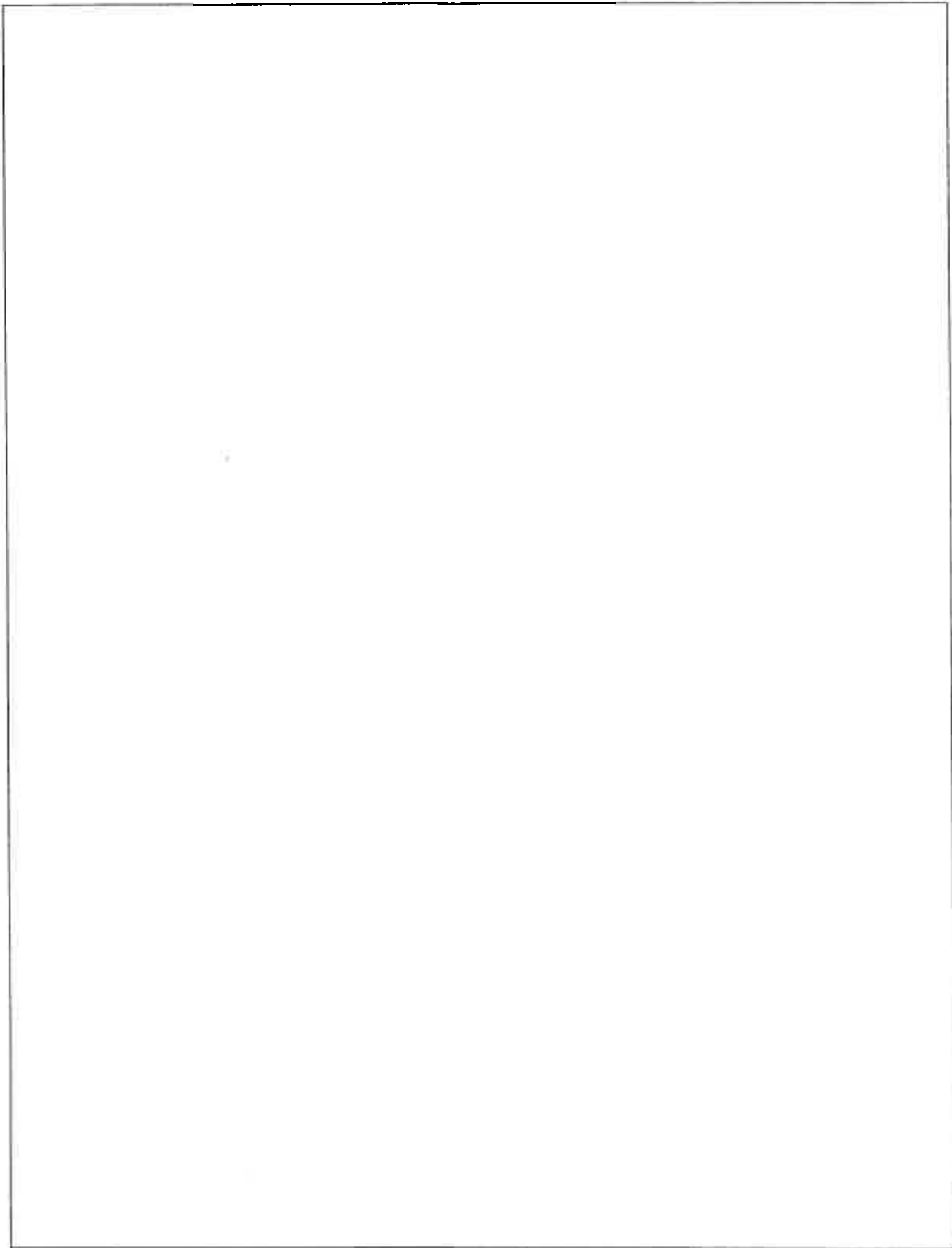
    int temp;
```

```
// Your code should be inserted here...
```

```
cout << "The height of the players in descending order:";
for(int i=0; i<5; i++)
    cout << height[i] << " ";
cout << "\nThe tallest player is "<< height[0] << " cm. ";
}
```

Continued...

- c. Draw the flowchart for a program that lists all the numbers between 0 and 100 that are divisible by 6 and outputs the sum of those numbers. [3 marks]



**Continued...**

- d. Based on the above flowchart, write the complete program in C++. [2 marks]

### **Question 3**

- a. Given the array, implement the following instructions in C++. [2 marks]

```
int abc[6] = {2, 3, 4, 5, 6, 7};
```

- Declare TWO (2) pointers p1 and p2 and point p1 to first element of the array abc and p2 to the second element of array abc.
- Use the pointers (p1 and p2) to calculate the sum of the first two elements of the array.

- b. Elaborate on the concept of function overloading and provide an example of function overloading to further justify your answer. [2 marks]

**Continued...**

- c. Write the C++ definition of the following functions in order to get the user's height and weight, calculate the Body Mass Index (BMI) and print the interpretation of the BMI.

[3 marks]

- i. Function "*calculate\_bmi*" takes the height and weight of the user as its arguments and calculates and returns the user's BMI. The formula for calculating BMI is as follows.

$$\text{BMI} = \frac{\text{Weight}}{\text{Height}^2} \text{ (Weight in Kilograms and Height in Meters)}$$

- ii. Function "*interpret\_bmi*" takes the BMI of the user as its argument and outputs the interpretations of the BMI based on the following table. This function does not return any value.

| BMI            | Weight Status            |
|----------------|--------------------------|
| Below 18.5     | Underweight              |
| 18.5 – 24.9    | Normal or Healthy Weight |
| 25.0 – 29.9    | Overweight               |
| 30.0 and Above | Obese                    |

Continued...

- d. Write a complete function definition for function "*computation*" which is called in the main function below. The function stores the subtraction and the addition results of the two integer values (num1 and num2) in the reference parameters (add and sub).

[2 marks]

```
int main (){
    int num1, num2, add, sub;
    cout << "Enter the first number: ";
    cin >> num1;
    cout << "Enter the second number: ";
    cin >> num2;
    computation(num1, num2, add, sub);
    cout << "The addition is: " << add << endl;
    cout << "The      subtraction is: " << sub;
}
```

*Sample output:*

```
Enter the first number: 8
Enter the second number: 5
The addition is: 13
The subtraction is: 3
```

```
Process exited after 3.553 seconds with return value 0
Press any key to continue . . .
```

- e. What is the output produced by the following code segment?

[1 mark]

```
int fun1(int x){
    return x+2;
}
int fun2(int a, int b){
    return fun1(a) * b--;
}
int main (){
    int a = 1, b = 2, c = 3;
    a = fun1(c);
    cout << a << " " << b << " " << c << endl;
    c = fun2 (a, b++);
    cout << a << " " << b << " " << c << endl;
}
```

Continued...

**Question 4**

- a. What is the output of the program below after executing? Explain the output. [2 marks]

```
class First{
    protected: int a;
    public:
        First(){ a = 1; }
};

class Second: protected First{
    public:
        Second(){ a = 2; }
};

int main(){
    First s;
    cout<<"a = "<<s.a;
}
```

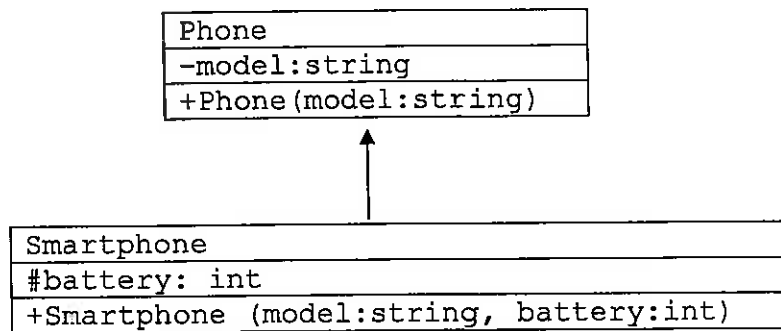
- b. Rewrite and transform the following class definition into an abstract class. [1 mark]

```
class Exam{
    public:
        void message() {cout<<"Good Luck";}
};
```

Continued...

c. Write the class definitions based on the Class Diagram below.

[4 marks]



d. Complete the definition of class Payment with the instructions below:

[3 marks]

- Overloaded constructor that initializes all attributes.
- Method that calculates the payment (hint:  $\text{pay} = \text{hour} * \text{rate}$ ).
- Destructor that displays a message: "Object is destroyed".

```
class Payment{
    double rate, hour;
    public:
        Payment(){
            rate = 0;  hour = 0;
        }
};
```

Continued...

**Question 5**

- a. Class AAA should allow only class BBB and global function printMessage to access its message. Complete the declaration for class AAA. [2 marks]

```
class AAA{  
    int message;
```

```
};
```

```
class BBB{  
    public:  
        void print (AAA a){ cout << a.message; }  
};
```

```
void printMessage (AAA a){  
    cout << a.message;  
}
```

**Continued...**

- b. Based on the definition of class One, complete the definition of class Two and class Three with the appropriate constructors. Class Two has a private integer TID, whereas class Three has a private integer ThID. [3 marks]

```
class One {  
    int OID;  
    public:  
    One (int OID): OID(OID) {}  
};  
  
class Two: public One{ . . . };  
  
class Three: public Two{ . . . };
```

- c. Rewrite the following program using exception handling. The program allows the user to enter the exam marks in the range 0 to 100. [3 marks]

```
int main(){  
    int mark;  
    cout<<"Input the marks:"<<endl;  
    cin>>mark;  
    if(mark>=0 && mark<=100)  
        cout<<"Your marks is "<<mark<<endl;  
    else  
        cout<<"Invalid input "<<endl;  
}
```

Continued...

d. Convert the following function to a template version.

[2 marks]

```
void printA (int a[], int size){  
    for (int i = 0; i < size; i++)  
        cout << a[i] << endl;  
}
```

**End of Paper**